Restricted maximum likelihood estimation for animal models using derivatives of the likelihood

K Meyer, Sp Smith

Original article

Restricted maximum likelihood estimation for animal models using derivatives of the likelihood

K Meyer, SP Smith *

Animal Genetics and Breeding Unit, University of New England, Armidale, NSW 2351, Australia

(Received 21 March 1995; accepted 9 October 1995)

Summary – Restricted maximum likelihood estimation using first and second derivatives of the likelihood is described. It relies on the calculation of derivatives without the need for large matrix inversion using an automatic differentiation procedure. In essence, this is an extension of the Cholesky factorisation of a matrix. A reparameterisation is used to transform the constrained optimisation problem imposed in estimating covariance components to an unconstrained problem, thus making the use of Newton–Raphson and related algorithms feasible. A numerical example is given to illustrate calculations. Several modified Newton–Raphson and method of scoring algorithms are compared for applications to analyses of beef cattle data, and contrasted to a derivative-free algorithm.

restricted maximum likelihood / derivative / algorithm / variance component estimation

Résumé – Estimation du maximum de vraisemblance restreinte pour des modèles individuels par dérivation de la vraisemblance. *Cet article décrit une méthode d'estimation du maximum de vraisemblance restreinte utilisant les dérivées première et seconde de la vraisemblance. La méthode est basée sur une procédure de différenciation automatique ne nécessitant pas l'inversion de grandes matrices. Elle constitue en fait une extension de la décomposition de Cholesky appliquée à une matrice. On utilise un paramétrage qui transforme le problème d'optimisation avec contrainte que soulève l'estimation des composantes de variance en un problème sans contrainte, ce qui rend possible l'utilisation d'algorithmes de Newton-Raphson ou apparentés. Les calculs sont illustrés sur un exemple numérique. Plusieurs algorithmes, de type Newton-Raphson ou selon la méthode des scores, sont appliqués à l'analyse de données sur bovins à viande. Ces algorithmes sont comparés entre eux et par ailleurs comparés à un algorithme sans dérivation.*

maximum de vraisemblance restreinte / dérivée / algorithme / estimation de composante de variance

* On leave from: EA Engineering, 3468 Mt Diablo Blvd, Suite B-100, Lafayette, CA 94549, USA

INTRODUCTION

Maximum likelihood estimation of (co)variance components generally requires the numerical solution of a constrained nonlinear optimisation problem (Harville, 1977). Procedures to locate the minimum or maximum of a function are classified according to the amount of information from derivatives of the function utilised; see, for instance, Gill et al (1981). Methods using both first and second derivatives are fastest to converge, often showing quadratic convergence, while search algorithms not relying on derivatives are generally slow, ie, require many iterations and function evaluations.

Early applications of restricted maximum likelihood (REML) estimation to animal breeding data used a Fisher's method of scoring type algorithm, following the original paper by Patterson and Thompson (1971) and Thompson (1973). This requires expected values of the second derivatives of the likelihood to be evaluated, which proved computationally highly demanding for all but the simplest analyses. Hence expectation–maximization (EM) type algorithms gained popularity and found widespread use for analyses fitting a sire model. Effectively, these use first derivatives of the likelihood function. Except for special cases, however, they required the inverse of a matrix of size equal to the number of random effects fitted, eg, number of sires times number of traits, which severely limited the size of analyses feasible.

For analyses under the animal model, Graser et al (1987) thus proposed a derivative-free algorithm. This only requires factorising the coefficient matrix of the mixed-model equations rather than inverting it, and can be implemented efficiently using sparse matrix techniques. Moreover, it is readily extendable to animal models including additional random effects and multivariate analyses (Meyer, 1989, 1991).

Multi-trait animal model analyses fitting additional random effects using a derivative-free algorithm have been shown to be feasible. However, they are computationally highly demanding, the number of likelihood evaluations required increasing exponentially with the number of (co)variance components to be estimated simultaneously. Groeneveld et al (1991), for instance, reported that 56 000 evaluations were required to reach a change in likelihood smaller than 10^{-7} when estimating 60 covariance components for five traits. While judicious choice of starting values and search strategies (eg, temporary maximisation with respect to a subset of the parameters only) together with exploitation of special features of the data structure might reduce demands markedly for individual analyses, it remains true that derivative-free maximisation in high dimensions is very slow to converge.

This makes a case for REML algorithms using derivatives of the likelihood for multivariate, multidimensional animal model analyses. Misztal (1994) recently presented a comparison of rates of convergence of derivative-free and derivative algorithms, concluding that the latter had the potential to be faster in almost all cases, in particular that their convergence rate depended little on the number of traits considered. Large-scale animal model applications using an EM type algorithm (Misztal, 1990) or even a method of scoring algorithm (Ducrocq, 1993) have been reported, obtaining the large matrix inverse (or its trace) required by the use of a supercomputer or applying some approximation. This paper describes

REML estimation under an animal model using first and second derivatives of the likelihood function, computed without inverting large matrices.

DERIVATIVES OF THE LIKELIHOOD

Consider the linear mixed model

$$\mathbf{y} = \mathbf{Xb} + \mathbf{Zu} + \mathbf{e} \qquad [1]$$

where $\mathbf{y}$, $\mathbf{b}$, $\mathbf{u}$ and $\mathbf{e}$ denote the vectors of observations, fixed effects, random effects and residual errors, respectively, and $\mathbf{X}$ and $\mathbf{Z}$ are the incidence matrices pertaining to $\mathbf{b}$ and $\mathbf{u}$. Let $V(\mathbf{u}) = \mathbf{G}$, $V(\mathbf{e}) = \mathbf{R}$ and $\mathrm{Cov}(\mathbf{u}, \mathbf{e}') = \mathbf{0}$, so that $V(\mathbf{y}) = \mathbf{V} = \mathbf{ZGZ}' + \mathbf{R}$. Assuming a multivariate normal distribution, ie, $\mathbf{y} \sim N(\mathbf{Xb}, \mathbf{V})$, the log of the REML likelihood ($\mathcal{L}$) is (eg, Harville, 1977)

$$\log \mathcal{L} = -\frac{1}{2}\left[\text{const} + \log|\mathbf{V}| + \log|\mathbf{X}^{*'}\mathbf{V}^{-1}\mathbf{X}^{*}| + (\mathbf{y} - \mathbf{X}\hat{\mathbf{b}})'\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\hat{\mathbf{b}})\right] \qquad [2]$$

where $\mathbf{X}^*$ denotes a full-rank submatrix of $\mathbf{X}$.

REML algorithms using derivatives have generally been derived by differentiating [2]. However, as outlined previously (Graser et al, 1987; Meyer, 1989), log $\mathcal{L}$ can be rewritten as

$$\log \mathcal{L} = -\frac{1}{2}\left[\text{const} + \log|\mathbf{R}| + \log|\mathbf{G}| + \log|\mathbf{C}| + \mathbf{y}'\mathbf{Py}\right] \qquad [3]$$

where $\mathbf{C}$ is the coefficient matrix in the mixed-model equations (MME) pertaining to [1] (or a full rank submatrix thereof), and $\mathbf{P}$ is a matrix,

$$\begin{aligned}\mathbf{P} &= \mathbf{V}^{-1} - \mathbf{V}^{-1}\mathbf{X}(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}\mathbf{X}'\mathbf{V}^{-1}\\ &= \mathbf{V}^{-1} - \mathbf{V}^{-1}\mathbf{X}^{*}(\mathbf{X}^{*\prime}\mathbf{V}^{-1}\mathbf{X}^{*})^{-1}\mathbf{X}^{*\prime}\mathbf{V}^{-1}\end{aligned}$$

Alternative forms of the derivatives of the likelihood can then be obtained by differentiating [3] instead of [2]. Let $\boldsymbol{\theta}$ denote the vector of parameters to be estimated with elements $\theta_i, i = 1, \ldots, p$. The first and second partial derivatives of the log likelihood are then

$$\frac{\partial \log \mathcal{L}}{\partial \theta_i} = -\frac{1}{2}\left[\frac{\partial \log|\mathbf{R}|}{\partial \theta_i} + \frac{\partial \log|\mathbf{G}|}{\partial \theta_i} + \frac{\partial \log|\mathbf{C}|}{\partial \theta_i} + \frac{\partial \mathbf{y}'\mathbf{Py}}{\partial \theta_i}\right] \qquad [4]$$

$$\frac{\partial^2 \log \mathcal{L}}{\partial \theta_i \partial \theta_j} = -\frac{1}{2}\left[\frac{\partial^2 \log|\mathbf{R}|}{\partial \theta_i \partial \theta_j} + \frac{\partial^2 \log|\mathbf{G}|}{\partial \theta_i \partial \theta_j} + \frac{\partial^2 \log|\mathbf{C}|}{\partial \theta_i \partial \theta_j} + \frac{\partial^2 \mathbf{y}'\mathbf{Py}}{\partial \theta_i \partial \theta_j}\right] \qquad [5]$$

Graser et al (1987) show how the last two terms in [3], $\log|\mathbf{C}|$ and $\mathbf{y}'\mathbf{Py}$, can be evaluated in a general way for all models of form [1] by carrying out a series of Gaussian elimination steps on the coefficient matrix in the MME augmented by the vector of right-hand sides and a quadratic in the data vector. Depending on the model of analysis and structure of $\mathbf{G}$ and $\mathbf{R}$, the other two terms required in [3], $\log|\mathbf{G}|$ and $\log|\mathbf{R}|$, can usually be obtained indirectly as outlined by Meyer (1989, 1991), generally requiring only matrix operations proportional to the number of traits considered. Derivatives of these four terms can be evaluated analogously.

Calculating $log|\mathbf{C}|$ *and* $\mathbf{y'Py}$ *and their derivatives*

The mixed-model matrix (MMM) or augmented coefficient matrix pertaining to [1] is

$$\mathbf{M} = \begin{bmatrix} \mathbf{X'R^{-1}X} & \mathbf{X'R^{-1}Z} & \mathbf{X'R^{-1}y} \\ \mathbf{Z'R^{-1}X} & \mathbf{Z'R^{-1}Z + G^{-1}} & \mathbf{Z'R^{-1}y} \\ \mathbf{y'R^{-1}X} & \mathbf{y'R^{-1}Z} & \mathbf{y'R^{-1}y} \end{bmatrix} = \begin{bmatrix} \mathbf{C} & \mathbf{r} \\ \mathbf{r'} & \mathbf{y'R^{-1}y} \end{bmatrix} \qquad [6]$$

where $\mathbf{r}$ is the vector of right-hand sides in the MME.

Using general matrix results, the derivatives of $\log|\mathbf{C}|$ are

$$\frac{\partial \log|\mathbf{C}|}{\partial \theta_i} = \operatorname{tr}\left(\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \theta_i}\right) \qquad [7]$$

$$\frac{\partial^2 \log|\mathbf{C}|}{\partial \theta_i \partial \theta_j} = \operatorname{tr}\left(\mathbf{C}^{-1}\frac{\partial^2 \mathbf{C}}{\partial \theta_i \partial \theta_j}\right) - \operatorname{tr}\left(\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \theta_i}\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \theta_j}\right) \qquad [8]$$

Partitioned matrix results give $\log|\mathbf{M}| = \log|\mathbf{C}| + \log(\mathbf{y'Py})$, ie, (Smith, 1995)

$$\mathbf{y'Py} = |\mathbf{M}|/|\mathbf{C}| \qquad [9]$$

This gives derivatives

$$\frac{\partial \mathbf{y'Py}}{\partial \theta_i} = \mathbf{y'Py}\left[\operatorname{tr}\left(\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \theta_i}\right) - \operatorname{tr}\left(\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \theta_i}\right)\right] \qquad [10]$$

$$\begin{aligned} \frac{\partial^2 \mathbf{y'Py}}{\partial \theta_i \partial \theta_j} = \mathbf{y'Py}\Bigg(&\left[\operatorname{tr}\left(\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \theta_i}\right) - \operatorname{tr}\left(\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \theta_i}\right)\right]\left[\operatorname{tr}\left(\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \theta_j}\right) - \operatorname{tr}\left(\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \theta_j}\right)\right] \\ &+ \operatorname{tr}\left(\mathbf{M}^{-1}\frac{\partial^2 \mathbf{M}}{\partial \theta_i \partial \theta_j}\right) - \operatorname{tr}\left(\mathbf{C}^{-1}\frac{\partial^2 \mathbf{C}}{\partial \theta_i \partial \theta_j}\right) \\ &- \operatorname{tr}\left(\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \theta_i}\mathbf{M}^{-1}\frac{\partial \mathbf{M}}{\partial \theta_j}\right) + \operatorname{tr}\left(\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \theta_i}\mathbf{C}^{-1}\frac{\partial \mathbf{C}}{\partial \theta_j}\right)\Bigg) \end{aligned} \qquad [11]$$

Obviously, these expressions ([7], [8], [10] and [11]) involving the inverse of the large matrices $\mathbf{M}$ and $\mathbf{C}$ are computationally intractable for any sizable animal model analysis. However, the Gaussian elimination procedure with diagonal pivoting advocated by Graser et al (1987) is only one of several ways to 'factor' a matrix. An alternative is a Cholesky decomposition. This lends itself readily to the solution of large positive definite systems of linear equations using sparse matrix storage schemes. Appropriate Fortran routines are given, for instance, by George and Liu (1981) and have been used successfully in derivative-free REML applications instead of Gaussian elimination (Boldman and Van Vleck, 1991).

The Cholesky decomposition factors a positive definite matrix into the product of a lower triangular matrix and its transpose. Let $\mathbf{L}$ with elements l_{ij} ($l_{ij} = 0$

for $j > i$) denote the Cholesky factor of $\mathbf{M}$, ie, $\mathbf{M} = \mathbf{LL}'$. The determinant of a triangular matrix is simply the product of its diagonal elements. Hence, with M denoting the size of $\mathbf{M}$,

$$\log|\mathbf{M}| = 2\sum_{k=1}^{M} \log l_{kk} \qquad [12]$$

$$\log|\mathbf{C}| = 2\sum_{k=1}^{M-1} \log l_{kk} \qquad [13]$$

and from [9],

$$\mathbf{y}'\mathbf{Py} = l_{MM}^2 \qquad [14]$$

Smith (1995) describes algorithms, outlined below, which allow the derivatives of the Cholesky factor of a matrix to be evaluated while carrying out the factorisation, provided the derivatives of the original matrix are specified.

Differentiating [13] and [14] then gives the derivatives of $\log|\mathbf{C}|$ and $\mathbf{y}'\mathbf{Py}$ as simple functions of the diagonal elements of the Cholesky matrix and its derivatives.

$$\frac{\partial \log|\mathbf{C}|}{\partial\theta_i} = 2\sum_{k=1}^{M-1} l_{kk}^{-1}\frac{\partial l_{kk}}{\partial\theta_i} \qquad [15]$$

$$\frac{\partial^2 \log|\mathbf{C}|}{\partial\theta_i\partial\theta_j} = 2\sum_{k=1}^{M-1} l_{kk}^{-1}\frac{\partial^2 l_{kk}}{\partial\theta_i\partial\theta_j} - l_{kk}^{-2}\frac{\partial l_{kk}}{\partial\theta_i}\frac{\partial l_{kk}}{\partial\theta_j} \qquad [16]$$

$$\frac{\partial \mathbf{y}'\mathbf{Py}}{\partial\theta_i} = 2l_{MM}\frac{\partial l_{MM}}{\partial\theta_i} \qquad [17]$$

$$\frac{\partial^2 \mathbf{y}'\mathbf{Py}}{\partial\theta_i\partial\theta_j} = 2\left(l_{MM}\frac{\partial^2 l_{MM}}{\partial\theta_i\partial\theta_j} + \frac{\partial l_{MM}}{\partial\theta_i}\frac{\partial l_{MM}}{\partial\theta_j}\right) \qquad [18]$$

Calculating** log|**R**| **and its derivatives

Consider a multivariate analysis for q traits and let $\mathbf{y}$ be ordered according to traits within animals. Assuming that error covariances between measurements on different animals are zero, $\mathbf{R}$ is blockdiagonal for animals,

$$\mathbf{R} = \sum_{i=1}^{N}{}^{+}\mathbf{R}_i \qquad [19]$$

where is N the number of animals which have records, and $\sum^{+}$ denotes the direct matrix sum (Searle, 1982). Hence $\log|\mathbf{R}|$ as well as its derivatives can be determined by considering one animal at a time.

Let $\mathbf{E}$ with elements e_{ij} ($i \leqslant j = 1, ..., q$) be the symmetric matrix of residual or error covariances between traits. For q traits, there are a total of $W = 2^q - 1$ possible combinations of traits recorded (assuming single records per trait), eg, $W = 3$ for $q = 2$ with combinations trait 1 only, trait 2 only and both traits. For animal i which has combination of traits w, $\mathbf{R}_i$ is equal to $\mathbf{E}_w$, the submatrix of $\mathbf{E}$ obtained by deleting rows and columns pertaining to missing records. As outlined by Meyer (1991), this gives

$$\log|\mathbf{R}| = \sum_{w=1}^{W} N_w \log|\mathbf{E}_w| \tag{20}$$

where N_w represents the number of animals having records for combination of traits w. Correspondingly,

$$\frac{\partial \log|\mathbf{R}|}{\partial\theta_i} = \sum_{w=1}^{W} N_w \frac{\partial \log|\mathbf{E}_w|}{\partial\theta_i} \tag{21}$$

$$\frac{\partial^2 \log|\mathbf{R}|}{\partial\theta_i\partial\theta_j} = \sum_{w=1}^{W} N_w \frac{\partial^2 \log|\mathbf{E}_w|}{\partial\theta_i\partial\theta_j} \tag{22}$$

Consider the case where the parameters to be estimated are the (co)variance components due to random effects and residual errors (rather than, for example, heritabilities and correlations), so that $\mathbf{V}$ is linear in $\boldsymbol{\theta}$, ie, $\mathbf{V} = \sum_{i=1}^{p} \theta_i \partial\mathbf{V}/\partial\theta_i$. Defining

$$\mathbf{D}_w^{\theta_i} = \frac{\partial \mathbf{E}_w}{\partial\theta_i}$$

with elements $d_{kl} = 1$, if the klth element of $\mathbf{E}_w$ is equal to θ_i, and $d_{kl} = 0$ otherwise, this then gives

$$\frac{\partial \log|\mathbf{R}|}{\partial\theta_i} = \sum_{w=1}^{W} N_w \mathrm{tr}(\mathbf{E}_w^{-1}\mathbf{D}_w^{\theta_i}) \tag{23}$$

$$\frac{\partial^2 \log|\mathbf{R}|}{\partial\theta_i\partial\theta_j} = -\sum_{w=1}^{W} N_w \mathrm{tr}(\mathbf{E}_w^{-1}\mathbf{D}_w^{\theta_i}\mathbf{E}_w^{-1}\mathbf{D}_w^{\theta_j}) \tag{24}$$

Let e_w^{rs} denote the rsth element of $\mathbf{E}_w^{-1}$. For $\theta_i = e_{kl}$ and $\theta_j = e_{mn}$, [23] and [24] then simplify to

$$\frac{\partial \log|\mathbf{R}|}{\partial\theta_i} = \sum_{w=1}^{W} N_w(2-\delta_{kl})e_w^{kl} \tag{25}$$

$$\frac{\partial^2 \log|\mathbf{R}|}{\partial\theta_i\partial\theta_j} = -\frac{1}{2}\sum_{w=1}^{W} N_w(2-\delta_{kl})(2-\delta_{mn})(e_w^{km}e_w^{ln} + e_w^{lm}e_w^{kn}) \tag{26}$$

where δ_{rs} is Kronecker's Delta, ie, $\delta_{rs} = 1$ for $r = s$ and zero otherwise. All other derivatives of $\log|\mathbf{R}|$ (ie, for θ_i or θ_j not equal to a residual covariance) are zero.

For $q = 1$ and $\mathbf{R} = \sigma_E^2\mathbf{I}$, [25] and [26] become $N\sigma_E^{-2}$ and $-N\sigma_E^{-4}$, respectively (for $\theta_i = \theta_j = \sigma_E^2$). Extensions for models with repeated records are straightforward. Hence, once the inverses of the matrices of residual covariances for all combination of numbers of traits recorded occurring in the data have been obtained (of maximum size equal to the maximum number of traits recorded per animal, and also required to set up the MMM), evaluation of $\log|\mathbf{R}|$ and its derivatives requires only scalar manipulations in addition.

Calculating log|G| and its derivatives

Terms arising from the covariance matrix of random effects, $\mathbf{G}$, can often be determined in a similar way, exploiting the structure of $\mathbf{G}$. This depends on the random effects fitted. Meyer (1989, 1991) describes $\log|\mathbf{G}|$ for various cases.

Define $\mathbf{T}$ with elements t_{ij} of size $rq \times rq$ as the matrix of covariances between random effects where r is the number of random factors in the model (excluding $\mathbf{e}$). For illustration, let $\mathbf{u}$ consist of a vector of animal genetic effects $\mathbf{a}$ and some uncorrelated additional random effect(s) $\mathbf{c}$ with N_C levels per trait, ie, $\mathbf{u}' = (\mathbf{a}'\mathbf{c}')$. In the simplest case, $\mathbf{a}$ consists of the direct additive genetic effects for each animal and trait, ie, it has length qN_A where N_A denotes the total number of animals in the analysis, including parents without records. In other cases, $\mathbf{a}$ might include a second genetic effect for each animal and trait, such as a maternal additive genetic effect, which may be correlated to the direct genetic effects. An example for $\mathbf{c}$ is a common environmental effect such as a litter effect.

With $\mathbf{a}$ and $\mathbf{c}$ uncorrelated, $\mathbf{T}$ can be partitioned into corresponding diagonal blocks $\mathbf{T}_A$ and $\mathbf{T}_C$, so that

$$\mathbf{G} = \text{Diag}\left\{\mathbf{A} \times \mathbf{T}_A; \mathbf{F} \times \mathbf{T}_C\right\} \qquad [27]$$

where $\mathbf{A}$ is the numerator relationship between animals, $\mathbf{F}$, often assumed to be the identity matrix, describes the correlation structure amongst the levels of $\mathbf{c}$, and × denotes the direct matrix product (Searle, 1982). This gives (Meyer, 1991)

$$\log|\mathbf{G}| = N_A \log|\mathbf{T_A}| + N_C \log|\mathbf{T_C}| + q(\log|\mathbf{A}| + \log|\mathbf{F}|) \qquad [28]$$

Noting that all $\partial^2\mathbf{T}/\partial\theta_i\partial\theta_j = 0$ (for $\mathbf{V}$ linear in $\boldsymbol{\theta}$), derivatives are

$$\frac{\partial \log|\mathbf{G}|}{\partial\theta_i} = N_A\text{tr}(\mathbf{T}_A^{-1}\mathbf{D}_A^{\theta_i}) + N_C\text{tr}(\mathbf{T}_C^{-1}\mathbf{D}_C^{\theta_i}) \qquad [29]$$

$$\frac{\partial^2 \log|\mathbf{G}|}{\partial\theta_i\partial\theta_j} = -N_A\text{tr}(\mathbf{T}_A^{-1}\mathbf{D}_A^{\theta_i}\mathbf{T}_A^{-1}\mathbf{D}_A^{\theta_j}) - N_C\text{tr}(\mathbf{T}_C^{-1}\mathbf{D}_C^{\theta_i}\mathbf{T}_C^{-1}\mathbf{D}_C^{\theta_j}) \qquad [30]$$

where $\mathbf{D}_A^{\theta_i} = \partial\mathbf{T}_A/\partial\theta_i$ and $\mathbf{D}_C^{\theta_i} = \partial\mathbf{T}_C/\partial\theta_i$ are again matrices with elements 1 if $t_{kl} = \theta_i$ and zero otherwise. As above, all second derivatives for θ_i and θ_j not pertaining to the same random factor (eg, $\mathbf{c}$) or two correlated factors (such as direct and maternal genetic effects) are zero. Furthermore, all derivatives of $\log|\mathbf{G}|$ with respect to residual covariance components are zero.

Further simplifications analogous to [25] and [26] can be derived. For instance, for a simple animal model fitting animals' direct additive genetic effects only as random effects ($r = 1$), $\mathbf{T}$ is the matrix of additive genetic covariances α_{ij} with $i, j = 1, \ldots, q$. For $\theta_i = \alpha_{kl}$ and $\theta_j = \alpha_{mn}$, this gives

$$\frac{\partial \log|\mathbf{G}|}{\partial \theta_i} = N_A(2-\delta_{kl})\alpha^{kl} \qquad [31]$$

$$\frac{\partial^2 \log|\mathbf{G}|}{\partial \theta_i \partial \theta_j} = -\frac{1}{2} N_A(2-\delta_{kl})(2-\delta_{mn})(\alpha^{km}\alpha^{ln} + \alpha^{lm}\alpha^{kn}) \qquad [32]$$

with α^{rs} denoting the rsth element of $\mathbf{T}^{-1}$. For $q = 1$ and $\alpha_{11} = \sigma_A^2$, [31] and [32] reduce to $N_A\sigma_A^{-2}$ and $-N_A\sigma_A^{-4}$, respectively.

Derivatives of the mixed model matrix

As emphasised above, calculation of the derivatives of the Cholesky factor of $\mathbf{M}$ requires the corresponding derivatives of $\mathbf{M}$ to be evaluated. Fortunately, these have the same structure as $\mathbf{M}$ and can be evaluated while setting up $\mathbf{M}$, replacing $\mathbf{G}$ and $\mathbf{R}$ by their derivatives.

For θ_i and θ_j equal to residual (co)variances, the derivatives of $\mathbf{M}$ are of the form

$$\begin{bmatrix} \mathbf{X}'\mathbf{Q}_R\mathbf{X} & \mathbf{X}'\mathbf{Q}_R\mathbf{Z} & \mathbf{X}'\mathbf{Q}_R\mathbf{y} \\ \mathbf{Z}'\mathbf{Q}_R\mathbf{X} & \mathbf{Z}'\mathbf{Q}_R\mathbf{Z} & \mathbf{Z}'\mathbf{Q}_R\mathbf{y} \\ \mathbf{y}'\mathbf{Q}_R\mathbf{X} & \mathbf{y}'\mathbf{Q}_R\mathbf{Z} & \mathbf{y}'\mathbf{Q}_R\mathbf{y} \end{bmatrix} \qquad [33]$$

with $\mathbf{Q}_R$ standing in turn for

$$\mathbf{Q}_{R_1} = \frac{\partial \mathbf{R}^{-1}}{\partial \theta_i} = -\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial \theta_i}\mathbf{R}^{-1} \qquad [34]$$

and

$$\mathbf{Q}_{R_2} = \frac{\partial^2 \mathbf{R}^{-1}}{\partial \theta_i \partial \theta_j} = \mathbf{R}^{-1}\left[\frac{\partial \mathbf{R}}{\partial \theta_i}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial \theta_j} + \frac{\partial \mathbf{R}}{\partial \theta_j}\mathbf{R}^{-1}\frac{\partial \mathbf{R}}{\partial \theta_i} - \frac{\partial^2 \mathbf{R}}{\partial \theta_i \partial \theta_j}\right]\mathbf{R}^{-1} \qquad [35]$$

for first and second derivatives, respectively. As outlined above, $\mathbf{R}$ is blockdiagonal for animals with submatrices $\mathbf{E}_w$. Hence, matrices $\mathbf{Q}_R$ have the same structure with submatrices

$$\mathbf{Q}_{R1_w} = -\mathbf{E}_w^{-1}\mathbf{D}_w^{\theta_i}\mathbf{E}_w^{-1} \qquad [36]$$

and (for $\mathbf{V}$ linear in $\boldsymbol{\theta}$ so that $\partial^2\mathbf{R}/\partial\theta_i\partial\theta_j = 0$)

$$\mathbf{Q}_{R2_w} = \mathbf{E}_w^{-1}\mathbf{D}_w^{\theta_i}\mathbf{E}_w^{-1}\mathbf{D}_w^{\theta_j}\mathbf{E}_w^{-1} + \mathbf{E}_w^{-1}\mathbf{D}_w^{\theta_j}\mathbf{E}_w^{-1}\mathbf{D}_w^{\theta_i}\mathbf{E}_w^{-1} \qquad [37]$$

Consequently, the derivatives of $\mathbf{M}$ with respect to the residual (co)variances can be set up in the same way as the 'data part' of $\mathbf{M}$. In addition to calculating the matrices $\mathbf{E}_w^{-1}$ for the W combination of records per animal occurring in the data, all derivatives of the $\mathbf{E}_w^{-1}$ for residual components need to evaluated. The extra calculations required, however, are trivial, requiring matrix operations proportional to the maximum number of records per animal only to obtain the terms in [36] and [37].

Analogously, for θ_i and θ_j equal to elements of $\mathbf{T}$, derivatives of $\mathbf{M}$ are

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_G & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \qquad [38]$$

with $\mathbf{Q}_G$ standing for

$$\mathbf{Q}_{G_1} = -\mathbf{G}^{-1}\frac{\partial \mathbf{G}}{\partial \theta_i}\mathbf{G}^{-1} \qquad [39]$$

for first derivatives, and

$$\mathbf{Q}_{G_2} = \mathbf{G}^{-1}\left(\frac{\partial \mathbf{G}}{\partial \theta_i}\mathbf{G}^{-1}\frac{\partial \mathbf{G}}{\partial \theta_j} + \frac{\partial \mathbf{G}}{\partial \theta_j}\mathbf{G}^{-1}\frac{\partial \mathbf{G}}{\partial \theta_i} - \frac{\partial^2 \mathbf{G}}{\partial \theta_i \theta_j}\right)\mathbf{G}^{-1} \qquad [40]$$

for second derivatives.

As above, further simplifications are possible depending on the structure of $\mathbf{G}$. For instance, for $\mathbf{G}$ as in [27] and $\partial^2\mathbf{G}/\partial\theta_i\partial\theta_j = 0$,

$$\mathbf{Q}_{G_1} = \begin{bmatrix} -\mathbf{T}_A^{-1}\mathbf{D}_A^{\theta_i}\mathbf{T}_A^{-1} \times \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & -\mathbf{T}_C^{-1}\mathbf{D}_C^{\theta_i}\mathbf{T}_C^{-1} \times \mathbf{D}^{-1} \end{bmatrix} \qquad [41]$$

and $\mathbf{Q}_{G_2} =$

$$\begin{bmatrix} \mathbf{T}_A^{-1}(\mathbf{D}_A^{\theta_i}\mathbf{T}_A^{-1}\mathbf{D}_A^{\theta_j} + \mathbf{D}_A^{\theta_j}\mathbf{T}_A^{-1}\mathbf{D}_A^{\theta_i})\mathbf{T}_A^{-1} \times \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{T}_C^{-1}(\mathbf{D}_C^{\theta_i}\mathbf{T}_C^{-1}\mathbf{D}_C^{\theta_j} + \mathbf{D}_C^{\theta_j}\mathbf{T}_C^{-1}\mathbf{D}_C^{\theta_i})\mathbf{T}_C^{-1} \times \mathbf{D}^{-1} \end{bmatrix} \qquad [42]$$

Expected values of second derivatives of $\log\mathcal{L}$

Differentiating [2] gives second derivatives of $\log\mathcal{L}$

$$\frac{\partial^2 \log\mathcal{L}}{\partial\theta_i\partial\theta_j} = -\frac{1}{2}\left[\mathrm{tr}\left(\mathbf{P}\frac{\partial^2\mathbf{V}}{\partial\theta_i\partial\theta_j}\right) - \mathrm{tr}\left(\mathbf{P}\frac{\partial\mathbf{V}}{\partial\theta_i}\mathbf{P}\frac{\partial\mathbf{V}}{\partial\theta_j}\right) - \mathbf{y}'\mathbf{P}\left(\frac{\partial^2\mathbf{V}}{\partial\theta_i\partial\theta_j} - 2\frac{\partial\mathbf{V}}{\partial\theta_i}\mathbf{P}\frac{\partial\mathbf{V}}{\partial\theta_j}\right)\mathbf{P}\mathbf{y}\right] \qquad [43]$$

with expected values (Harville, 1977)

$$E\left[\frac{\partial^2 \log\mathcal{L}}{\partial\theta_i\partial\theta_j}\right] = -\frac{1}{2}\mathrm{tr}(\mathbf{P}\frac{\partial\mathbf{V}}{\partial\theta_i}\mathbf{P}\frac{\partial\mathbf{V}}{\partial\theta_j}) \qquad [44]$$

Again, for $\mathbf{V}$ linear in $\boldsymbol{\theta}$, $\partial^2\mathbf{V}/\partial\theta_i\partial\theta_j = 0$. From [5] and noting that $\partial\mathbf{P}/\partial\theta_i = -\mathbf{P}(\partial\mathbf{V}/\partial\theta_i)\mathbf{P}$, ie, that the last term in [43] is the second derivative of $\mathbf{y}'\mathbf{P}\mathbf{y}$,

$$\begin{aligned} E[\frac{\partial^2 \log\mathcal{L}}{\partial\theta_i\partial\theta_j}] &= -\frac{\partial^2 \log\mathcal{L}}{\partial\theta_i\partial\theta_j} - \frac{1}{2}\frac{\partial^2\mathbf{y}'\mathbf{P}\mathbf{y}}{\partial\theta_i\partial\theta_j} \\ &= \frac{1}{2}\left[\frac{\partial^2\log|\mathbf{R}|}{\partial\theta_i\partial\theta_j} + \frac{\partial^2\log|\mathbf{G}|}{\partial\theta_i\partial\theta_j} + \frac{\partial^2\log|\mathbf{C}|}{\partial\theta_i\partial\theta_j}\right] \end{aligned} \qquad [45]$$

Hence, expected values of the second derivatives are essentially (sign ignored) equal to the observed values minus the contribution from the data, and thus can be evaluated analogously. With second derivatives of $\mathbf{y}'\mathbf{P}\mathbf{y}$ not required, computational requirements are reduced somewhat as only the first $M-1$ rows of $\partial^2\mathbf{M}/\partial\theta_i\partial\theta_j$ need to be evaluated and factored.

AUTOMATIC DIFFERENTIATION

Calculation of the derivatives of the likelihood as described above relies on the fact that the derivatives of the Cholesky factor of a matrix can be obtained 'automatically', provided the derivatives of the original matrix can be specified.

Smith (1995) describes a so-called forward differentiation, which is a straightforward expansion of the recursions employed in the Cholesky factorisation of a matrix $\mathbf{M}$. Operations to determine the latter are typically carried out sequentially by rows. Let $\mathbf{L}$, of size N, be initialised to $\mathbf{M}$. First, the pivot (diagonal element which must be greater than an operational zero) is selected for the current row k. Secondly, the off-diagonal elements for the row ('lead column') are adjusted (L_{jk} for $j = k+1, \ldots, N$), and thirdly the elements in the remaining part of $\mathbf{L}$ (L_{ij} for $j = k+1, \ldots, N$ and $i = j, \ldots, N$) are modified ('row operations'). After all N rows have been processed, $\mathbf{L}$ contains the Cholesky factor of $\mathbf{M}$.

Pseudo-code given by Smith (1995) for the calculation of the Cholesky factor and its first and second derivatives is summarised in table I. It can be seen that the operations to evaluate a second derivative require the respective elements of the two corresponding first derivatives. This imposes severe constraints on the memory requirements of the algorithm. While it is most efficient to evaluate the Cholesky factor and all its derivatives together, considerable space can be saved by computing the second derivatives one at a time. This can be done by holding all the first derivatives in memory, or, if core space is the limiting factor, storing first derivatives on disk (after evaluating them individually as well) and reading in only the two required. Hence, the minimum memory requirement for REML using first and second derivatives is $4 \times \mathbf{L}$, compared to $\mathbf{L}$ for a derivative-free algorithm.

Smith (1995) stated that, using forward differentiation, each first derivative required not more than twice the work required to evaluate log $\mathcal{L}$ only, and that the work needed to determine a second derivative would be at most four times that to calculate log $\mathcal{L}$.

In addition, Smith (1995) described a 'backward differentiation' scheme, so named because it reverses the order of steps in the forward differentiation. It is applicable for cases where we want to evaluate a scalar function of $\mathbf{L}$, $f(\mathbf{L})$, in our case $log|\mathbf{C}| + \mathbf{y}'\mathbf{P}\mathbf{y}$ which is a function of the diagonal elements of $\mathbf{L}$ (see [13] and [14]). It requires computing a (lower triangular) matrix $\mathbf{W}$ which, on completion of the backward differentiation, contains the derivatives of $f(\mathbf{L})$ with respect to the elements of $\mathbf{M}$. First derivatives of $f(\mathbf{L})$ can then be evaluated one at a time as $\mathrm{tr}(\mathbf{W}\partial\mathbf{M}/\partial\theta_r)$.

The pseudo-code given by Smith (1995) for the backward differentiation is shown in table II. Calculation of $\mathbf{W}$ requires about twice as much work as one likelihood evaluation, and, once $\mathbf{W}$ is evaluated, calculating individual derivatives (step 3 in table II) is computationally trivial, ie, evaluation of all first derivatives by backward

Table I. Pseudo-code for automatic 'forward differentiation' of a matrix M[a] given by Smith (1995).

Step			*Cholesky factor*	*1st derivatives*	*2nd derivatives*
1		Initialise $i, j = 1, \ldots, N$	$L_{ij} := M_{ij}$	$L'_{(r)ij} := \partial M_{ij} / \partial \theta_r$	$L''_{(rs)ij} := \partial^2 M_{ij} / \partial \theta_r \partial \theta_s$
2	a	Pivot $k = 1, \ldots, N$	$L_{kk} := \sqrt{L_{kk}}$	$L'_{(r)kk} := 0.5 L'_{(r)\boldsymbol{kk}} / L_{kk}$	$L''_{(rs)kk} := [0.5 L''_{(rs)kk} - L'_{(r)kk} L'_{(s)kk}] / L_{kk}$
	b	Lead column $j = k + 1, \ldots, N$	$Ljk := L_{jk} / L_{kk}$	$L'_{(r)jk} := [L'_{(r)jk} -$ $L_{jk} L'_{(r)kk}] / L_{kk}$	$L''_{(rs)jk} := [L''_{(rs)jk} - L_{jk} L''_{(rs)kk}$ $- L'_{(r)jk} L'_{(s)kk} - L'_{(s)jk} L'_{(r)kk}] / L_{kk}$
	c	Row operations $j = k + 1, \ldots, N$ $i = j, \ldots, N$	$L_{ij} := L_{ij} - L_{ik} L_{jk}$	$L'_{(r)ij} := L'_{(r)ij}$ $- L'_{(r)ik} L_{jk} - L_{\boldsymbol{ik}} L'_{(r)jk}$	$L''_{(rs)ij} := L''_{(rs)ij} - L''_{(rs)\boldsymbol{ik}} L_{jk} - L_{ik} L''_{(rs)jk}$ $- L'_{(r)ik} L'_{(s)jk} - L'_{(s)ik} L'_{(r)jk}$

[a] Assumed to be positive definite so that all $L_{kk} > 0$.

differentiation requires only somewhat more work than calculation of one derivative by forward differentiation. Smith (1995) also described the calculation of second derivatives by backward differentiation (pseudo-code not shown here). Amongst other calculations, this involves one evaluation of a matrix $\mathbf{W}$ as described above, for each parameter and requires another work array of size $\mathbf{L}$ in addition to space to store at least one matrix of derivatives of $\mathbf{M}$. Hence the minimum memory requirement for this algorithm is $3 \times \mathbf{L} + \mathbf{M}$ ($\mathbf{M}$ and $\mathbf{L}$ differing by the fill-in created during the factorisation). Smith (1995) claimed that the total work required to evaluate all second derivatives for p parameters was no more than $6p$ times that for a likelihood evaluation.

Table II. Pseudo-code for automatic 'forward differentiation' and calculation of first derivatives of a matrix $\mathbf{M}$[a], given by Smith (1995).

Step				
1		Initialise $i \leqslant j = 1, \ldots, N$	$W_{ij} := \partial f(\mathbf{L})/\partial L_{ij}$	
2	a	Row operations $j = k+1, \ldots, N$ $i = j, \ldots, N$	$W_{ik} := W_{ik} - W_{ij} L_{jk}$	$W_{jk} := W_{jk} - W_{ij} L_{ik}$
	b	Lead column $j = k+1, \ldots, N$	$W_{jk} := W_{jk}/L_{kk}$	$W_{kk} := W_{kk} - L_{jk} W_{jk}$
	c	Pivot $k = N, \ldots, 1$	$W_{kk} := 0.5 W_{kk}/L_{kk}$	
3		1st derivatives	$\partial f(\mathbf{L})/\partial \theta_r = \sum_{i=1}^{N} \sum_{j=1}^{i} W_{ij} \partial M_{ij}/\partial \theta_r$	

[a] Assumed to be positive definite so that all $L_{kk} > 0$. [b] Assume elements of the Cholesky factor, L_{ij}, have already been calculated.

MAXIMISING THE LIKELIHOOD

Methods to locate the maximum of the likelihood function in the context of variance component estimation are reviewed, for instance, by Harville (1977) and Searle et al (1992; Chapter 8). Most utilise the gradient vector, ie, vector of first derivatives of the likelihood function, to determine the direction of search.

Using second derivatives

One of the oldest and most widely used methods to optimise a non-linear function is the Newton–Raphson (NR) algorithm. It requires the Hessian matrix of the function, ie, the matrix of second partial derivatives of the (log) likelihood with respect to the parameters to be estimated. Let $\boldsymbol{\theta}^t$ denote the estimate of $\boldsymbol{\theta}$ at the tth round of iteration. The next estimate is then obtained as

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - (\mathbf{H}^t)^{-1} \mathbf{g}^t \qquad [46]$$

where $\mathbf{H}^t = \{\partial^2 \log \mathcal{L}/\partial\theta_i\partial\theta_j\}$ and $\mathbf{g}^t = \{\partial \log \mathcal{L}/\partial\theta_i\}$ are the Hessian matrix and gradient vector of $\log \mathcal{L}$, respectively, both evaluated at $\boldsymbol{\theta} = \boldsymbol{\theta}^t$. While the NR algorithm can be quick to converge, in particular for functions resembling a quadratic function, it is known to be sensitive to poor starting values (Powell, 1970). Unlike other algorithms, it is not guaranteed to converge though global convergence has been shown for some cases using iterative partial maximisation (Jensen et al, 1991).

In practice, so-called extended or modified NR algorithms have been found to be more successful. Jennrich and Sampson (1976) suggested step halving, applied successively until the likelihood is found to increase, to avoid 'overshooting'. More generally, the change in estimates for the tth iterate in [46] is given by

$$\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t = \mathbf{B}^t\mathbf{g}^t \qquad [47]$$

for the extended NR, $\mathbf{B}^t = -\tau^t(\mathbf{H}^t)^{-1}$, where τ^t is a step-size scaling factor. The optimum for τ^t can be determined readily as the value which results in the largest increase in likelihood, using a one-dimensional maximisation technique (Powell, 1970). This relies on the direction of search given by $\mathbf{H}^{-1}\mathbf{g}$ generally being a 'good' direction and that, for $-\mathbf{H}$ positive definite, there is always a step-size which will increase the likelihood.

Alternatively, the use of

$$\mathbf{B}^t = (\kappa^t\mathbf{I} - \mathbf{H}^t)^{-1} \qquad [48]$$

has been suggested (Marquardt, 1963) to improve the performance of the NR algorithm. This results in a step intermediate between a NR step ($\kappa = 0$) and a method of steepest ascent step (κ large). Again, κ can be chosen to maximise the increase in $\log \mathcal{L}$, though for large values of κ the step size is small, so that there is no need to include a search step in the iteration (Powell, 1970).

Often expected values of the second derivatives of $\log \mathcal{L}$ are easier to calculate than the observed values. Replacing $-\mathbf{H}$ by the information matrix

$$\mathbf{I}(\boldsymbol{\theta}) = \{E[\{\partial^2 \log \mathcal{L}/\partial\theta_i\partial\theta_j]\}$$

results in Fisher's method of scoring (MSC). It can be extended or modified in the same way as the NR algorithm (Harville, 1977). Jennrich and Sampson (1976) and Jennrich and Schluchter (1986) compared NR and MSC, showing that the MSC was generally more robust against a poor choice of starting values than the NR, though it tended to require more iterations. They thus recommended a scheme using the MSC initially and switching to NR after a few rounds of iteration when the increase in $\log \mathcal{L}$ between steps was less than one.

Using first derivatives only

Other methods, so-called variable-metric or Quasi-Newton procedures, essentially use the same strategies, but replace $\mathbf{B}$ by an approximation of the Hessian matrix. Often starting from the identity matrix, this is updated with each round of iteration, requiring only first derivatives of the likelihood function, and converges to the Hessian for sufficient number of iterations. A detailed review of these methods is given by Dennis and Moré (1977).

An interesting variation has recently been presented by Johnson and Thompson (1995). Noting that the observed and expected information were of opposite sign and differed only by a term involving the second derivatives of $\mathbf{y}'\mathbf{P}\mathbf{y}$ (see [5] and [45]), they suggested using the average of observed and expected information (AI) to approximate the Hessian matrix. Since it requires only the second derivatives of $\mathbf{y}'\mathbf{P}\mathbf{y}$ to be evaluated, each iterate is computationally considerably less demanding than a 'full' NR or MSC step, and the same modifications as described above for the NR (see [47] and [48]) can be applied. Initial comparisons of the rate of convergence and computer time required showed the AI algorithm to be highly advantageous over both derivative-free and EM-type procedures (Johnson and Thompson, 1995).

Constraining parameters

All the Newton-type algorithms described above perform an unconstrained optimisation. To estimating (co)variance components, however, we require variances be non-negative, correlations to be in the range from -1 to 1 and, for more than two traits, for them to be consistent with each other; more generally, estimated covariance matrices need to be (semi-) positive definite. As shown by Hill and Thompson (1978), the probability of obtaining parameter estimates out of bounds depends on the magnitude of the correlations (population values) and increases rapidly with the number of traits considered, in particular for genetic covariance matrices.

For univariate analyses, a common approach has been to set negative estimates of variance components to zero and continue iterating. Partial sweeping of $\mathbf{B}$ to handle boundary constraints, monitoring and restraining the size of pivots relative to the corresponding (original) diagonal elements has been recommended (Jennrich and Sampson, 1976; Jennrich and Schluchter, 1986). Harville (1977; section 6.3) distinguished between three types of techniques to modify unconstrained optimisation procedures to accommodate constraints on the parameter space. Firstly, penalty techniques operate on a function which is close to $\log \mathcal{L}$ except at the boundaries where it assumes large negative values which effectively serve as a barrier, deflecting a further search in that direction. Secondly, gradient projection techniques are suitable for linear inequality constraints. Thirdly, it may be feasible to transform the parameters to be estimated so that maximisation on the new scale is unconstrained.

Box (1965) demonstrated for several examples that the computational effort to solve constrained problems can be reduced markedly by eliminating constraints so that one of the more powerful unconstrained methods, preferably with quadratic convergence, can be employed. For univariate analysis, an obvious way to eliminate non-negativity constraints is to maximise $\log \mathcal{L}$ with respect to standard deviations and to square these on convergence (Harville, 1977). Alternatively, we could estimate logarithmic values of variance components instead of the variances. This seems preferable to taking square roots where, on backtransforming, a largish negative estimate might become a substantial positive estimate of the corresponding variance. Harville (1977) cautioned however that such transformations may result in the introduction of additional stationary points on the likelihood surface, and thus should be used only in conjunction with optimisation techniques ensuring an increase in $\log \mathcal{L}$ in each iteration.

Further motivation for the use of transformations has been provided by the scope to reduce computational effort or to improve convergence by making the shape of the likelihood function on the new scale more quadratic. For multivariate analyses with one random effect and equal design matrices, for instance, a canonical transformation allows estimation to be broken down into a series of corresponding univariate analyses (Meyer, 1985); see Jensen and Mao (1988) for a review. Harville and Callanan (1990) considered different forms of the likelihood for both NR and MSC and demonstrated how they affected convergence behaviour. In particular, a 'linearisation' was found to reduce the number of iterates required to reach convergence considerably. Thompson and Meyer (1986) showed how a reparameterisation aimed at making variables less correlated could speed up an expectation–maximisation algorithm dramatically.

For univariate analyses of repeated measures data with several random effects, Lindstrom and Bates (1988) suggested maximisation of $\log \mathcal{L}$ with respect to the non-zero elements of the Cholesky decomposition of the covariance matrix of random effects in order to remove constraints on the parameter space and to improve stability of the NR algorithm. In addition, they chose to estimate the error variance directly, operating on the profile likelihood of the remaining parameters. For several examples, the authors found consistent convergence of the NR algorithm when implemented this way, even for an overparameterised model. Recently, Groeneveld (1994) examined the effect of this reparameterisation for large-scale multivariate analyses using derivative-free REML, reporting substantial improvements in speed of convergence for both direct search (downhill Simplex) and Quasi-Newton algorithms.

IMPLEMENTATION

Reparameterisation

For our analysis, parameters to be estimated are the non-zero elements of the lower triangle of the two covariance matrices **E** and **T**, with **T** potentially consisting of independent diagonal blocks, depending on the random effects fitted. Let

$$\mathbf{U} = \text{Diag}\,\{\mathbf{T}; \mathbf{E}\} = \sum_{r=1}^{K} {}^{+}\mathbf{U}_r \qquad [49]$$

The Cholesky decomposition of **U** has the same structure

$$\mathbf{L}_U = \sum_{r=1}^{K} {}^{+}\mathbf{L}_{U_r} \qquad [50]$$

Estimating the non-zero elements of matrices $\mathbf{L}_{\mathbf{U}_\mathbf{r}}$ then ensures positive definite matrices $\mathbf{U}_\mathbf{r}$ on transforming back to the original scale (Lindstrom and Bates, 1988). However, for the $\mathbf{U}_r$ representing covariance matrices, the ith diagonal element of $\mathbf{L}_{U_r}$ can be interpreted as the conditional standard deviation of trait i (for random factor r) given traits 1 to $i-1$. Conceptually, this cannot be less than zero. Hence, it is suggested to apply a secondary transformation, estimating the logarithmic values of these diagonal elements and thus, as discussed above for univariate analyses of variance components, effectively forcing them to be greater than zero.

Let $\boldsymbol{\nu}$ denote the vector of parameters on the new scale. In order to maximise $\log \mathcal{L}$ with respect to the elements of $\boldsymbol{\nu}$, we need to transform its derivatives accordingly. Lindstrom and Bates (1988) describe briefly how to obtain the gradient vector and Hessian matrix for a Cholesky matrix $\mathbf{L}_{U_r}$ from those for the matrix $\mathbf{U}_r$ (see also corrections, Lindstrom and Bates (1994)).

For the ith element of $\boldsymbol{\nu}$,

$$\frac{\partial \log \mathcal{L}}{\partial \nu_i} = \sum_{k=1}^{p} \frac{\partial \theta_k}{\partial \nu_i} \frac{\partial \log \mathcal{L}}{\partial \theta_k} \qquad [51]$$

More generally, for a one-to-one transformation (Zacks, 1971)

$$\left\{ \frac{\partial \log \mathcal{L}}{\partial \nu_i} \right\} = \mathbf{J}' \left\{ \frac{\partial \log \mathcal{L}}{\partial \theta_i} \right\} \qquad [52]$$

where $\mathbf{J}$ with elements $\partial \theta_i / \partial \nu_j$ is the Jacobian of $\boldsymbol{\theta}$ with respect to $\boldsymbol{\nu}$.

Similarly,

$$\frac{\partial^2 \log \mathcal{L}}{\partial \nu_i \partial \nu_j} = \sum_{k=1}^{p} \frac{\partial^2 \theta_k}{\partial \nu_i \partial \nu_j} \frac{\partial \log \mathcal{L}}{\partial \theta_k} + \sum_{k=1}^{p} \sum_{m=1}^{p} \frac{\partial \theta_k}{\partial \nu_i} \frac{\partial \theta_m}{\partial \nu_j} \frac{\partial^2 \log \mathcal{L}}{\partial \theta_k \partial \theta_m} \qquad [53]$$

with the first part of [53] equal to the ith element of $(\partial \mathbf{J}' / \partial \nu_j) \{\partial \log \mathcal{L} / \partial \theta_k\}$ and the second part equal to ijth element of $\mathbf{J}' \{\partial^2 \log \mathcal{L} / \partial \theta_i \partial \theta_j\} \mathbf{J}$.

Consider one covariance matrix $\mathbf{U}_r$ at a time, dropping the subscript r for convenience, and let u_{st} and l_{wx} denote the elements of $\mathbf{U}$ and $\mathbf{L}_U$, respectively. From $\mathbf{U} = \mathbf{L}\mathbf{L}'$, it follows that

$$u_{st} = \sum_{k=1}^{\min(s,t)} l_{sk} l_{tk} \qquad [54]$$

where $\min(s,t)$ is the smaller value of s and t. Hence, the ijth element of $\mathbf{J}$ for $\theta_i = u_{st}$ and $\nu_j = l_{wx}$ is

$$\frac{\partial u_{st}}{\partial l_{wx}} = \sum_{k=1}^{\min(s,t)} \left(\frac{\partial l_{sk}}{\partial l_{wx}} l_{tk} + \frac{\partial l_{tk}}{\partial l_{wx}} l_{sk} \right) \qquad [55]$$

For $w \neq x$ and $s, t \geqslant x$, this is non-zero only if at least one of s and t is equal to w. Allowing for the log transformation of the diagonal elements (and using the fact that $\partial l_{mm} / \partial \log(l_{mm}) = l_{mm}$ for log values to the base e), this gives four different cases to consider :

$$\begin{aligned}\partial u_{wt}/\partial l_{wx} &= l_{tx}\\ \partial u_{ww}/\partial l_{wx} &= 2l_{wx}\\ \partial u_{wt}/\partial \log(l_{ww}) &= l_{tw}l_{ww}\\ \partial u_{ww}/\partial \log(l_{ww}) &= 2l_{ww}^2\end{aligned}$$

For example, for $q = 3$ traits and six elements in $\boldsymbol{\theta}$ (u_{11}, u_{12}, u_{13}, u_{22}, u_{23} and u_{33}) and $\boldsymbol{\nu}$ ($\log(l_{11})$, l_{21}, l_{31}, $\log(l_{22})$, l_{32}, $\log(l_{33})$)

$$\mathbf{J} = \begin{bmatrix} 2l_{11}^2 & 0 & 0 & 0 & 0 & 0\\ l_{21}l_{11} & l_{11} & 0 & 0 & 0 & 0\\ l_{31}l_{11} & 0 & l_{11} & 0 & 0 & 0\\ 0 & 2l_{21} & 0 & 2l_{22}^2 & 0 & 0\\ 0 & l_{31} & l_{21} & l_{32}l_{22} & l_{22} & 0\\ 0 & 0 & 2l_{31} & 0 & 2l_{32} & 2l_{33}^2 \end{bmatrix}$$

Similarly, the ijth element of $\{\partial^2\theta_k/\partial\nu_i\partial\nu_j\}$ (see [53]) for $\theta_k = u_{st}$, $\nu_i = l_{wx}$ and $\nu_j = l_{yz}$ is

$$\frac{\partial^2 u_{st}}{\partial l_{wx}\partial l_{yz}} = \sum_{k=1}^{\min(s,t)} \left(\frac{\partial l_{sk}}{\partial l_{wx}}\frac{\partial l_{tk}}{\partial l_{yz}} + \frac{\partial l_{tk}}{\partial l_{wx}}\frac{\partial l_{sk}}{\partial l_{yz}}\right) \qquad [56]$$

Allowing for the additional adjustments due to the log transformation of diagonals, [56] is non-zero in five cases (for $w \neq t \neq x$ and $x, t \leqslant w$):

$$\begin{aligned}\partial^2 u_{wt}/(\partial l_{wx}\partial l_{tx}) &= 1\\ \partial^2 u_{ww}/(\partial l_{wx})^2 &= 2\\ \partial^2 u_{wt}/(\partial \log(l_{ww})\partial l_{tw}) &= l_{ww}\\ \partial^2 u_{wt}/(\partial \log(l_{ww}))^2 &= l_{tw}l_{ww}\\ \partial^2 u_{ww}/(\partial \log(l_{ww}))^2 &= 4l_{ww}^2\end{aligned}$$

For the above example, this gives the first two derivatives of $\mathbf{J}$

$$\frac{\partial \mathbf{J}}{\partial \log(l_{11})} = \begin{bmatrix} 4l_{11}^2 & 0 & 0 & 0 & 0 & 0\\ l_{21}l_{11} & l_{11} & 0 & 0 & 0 & 0\\ l_{31}l_{11} & 0 & l_{11} & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & \end{bmatrix} \quad \text{and} \quad \frac{\partial \mathbf{J}}{\partial l_{21}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0\\ l_{11} & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 2 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

In some cases, a covariance component cannot be estimated, for instance, the error covariance between two traits measured on different sets of animals. On the variance component scale, this parameter is usually set to zero and simply not estimated. On the reparameterised (Cholesky) scale, however, the corresponding new parameter may be non-zero. This can be accommodated by, again, not estimating this parameter but calculating its new value from the estimates of the other parameters, similar to the way a correlation is fixed to a certain value by only estimating the respective variances and then calculating the new covariance 'estimate' from them.

For example, consider three traits with the covariance between traits 2 and 3, u_{23}, which are not estimable. Setting $u_{23} = 0$ gives a non-zero value on the reparameterised scale of $l_{32} = -l_{21}l_{31}/l_{22}$. Only the other five parameters $(l_{11}, l_{21}, l_{31}, l_{22}$, and $l_{33})$ are then estimated (ignoring the log transformation of diagonals here for simplicity) while an 'estimate' of l_{32} is calculated from those of l_{21}, l_{31} and l_{33} according to the relationship above. On transforming back to the original scale, this ensures that the covariance between traits 2 and 3 remains fixed at zero.

Sparse matrix storage

Calculation of $\log \mathcal{L}$ for use in derivative-free REML estimation under an animal model has been made feasible for practically sized data sets through the use of sparse matrix storage techniques. These adapt readily to include the calculation of derivatives of $\mathbf{L}$. One possibility is the use of so-called linked lists (see, for instance, Tier and Smith (1989)). George and Liu (1981) describe several other strategies, using a 'compressed' storage scheme when applying a Cholesky decomposition to a large symmetric, positive definite, sparse matrix.

Elements of the matrices of derivatives of $\mathbf{L}$ are subsets of elements of $\mathbf{L}$, ie, exist only for non-zero elements of $\mathbf{L}$. Thus the same system of pointers can be used for $\mathbf{L}$ and all its derivatives, reducing overheads for storage and calculation of addresses of individual elements (though at the expense of reserving space for zero derivatives corresponding to non-zero l_{ij}). Moreover, schemes aimed at reducing the 'fill-in' during the factorisation of $\mathbf{M}$, should also reduce computational requirements for determining derivatives.

Matrix storage required in evaluating derivatives of $\log \mathcal{L}$ can be considerable: for p parameters to be estimated, there are p first and $p(p+1)/2$ second derivatives, ie, up to $1+p(p+3)/2$ times as much space as for calculating $\log \mathcal{L}$ only can be required. Even for analyses with one random factor (animals) only, this becomes prohibitive very quickly, amounting to a factor of 28 for $q = 2$ traits and $p = 6$ parameters, and 91 for $q = 3$ and $p = 12$. However, while matrices $\partial\mathbf{L}/\partial\theta_i$ are needed to evaluate second derivatives, matrices $\partial^2\mathbf{L}/\partial\theta_i\partial\theta_j$ are not required after $\partial^2 \log|\mathbf{C}|/\partial\theta_i\partial\theta_j$ and $\partial^2\mathbf{y}'\mathbf{P}\mathbf{y}/\partial\theta_i\partial\theta_j$ have been calculated from their diagonal elements. Hence, while it is most efficient to evaluate all derivatives of each l_{ij} simultaneously, second derivatives can be determined one at a time after $\mathbf{L}$ and its first derivatives have been determined. This can be done by setting up and processing each $\partial^2\mathbf{M}/\partial\theta_i\partial\theta_j$ individually thus reducing memory required dramatically.

Software

Extended NR and MSC algorithms were implemented for the ten animal models accommodated by DFREML (Meyer, 1992), parameterising to elements of the

Cholesky decomposition of the covariance matrices (and logarithmic values of their diagonals), as described above, to remove constraints on the parameter space. Prior to estimation, the ordering of rows and columns 1 to $M - 1$ in $\mathbf{M}$ was determined using the minimum degree re-ordering performed by George and Liu's (1981) subroutine GENQMD, and their subroutine SBMFCT was used to establish the symbolic factorisation of $\mathbf{M}$ (all M rows and columns) and the associated compressed storage pointers, allocating space for all non-zero elements of $\mathbf{L}$. In addition, the use of the average information matrix was implemented. However, this was done merely for the comparison of convergence rates without making use of the fact that only the derivatives of $\mathbf{y}'\mathbf{P}\mathbf{y}$ were required.

For each iterate, the optimal step size or scaling factor (see [47] and [48]) was determined by carrying out a one-dimensional, derivative-free search. This was done using a quadratic approximation of the likelihood surface, allowing for up to five approximation steps per iterate. Other techniques, such as a simple step-halving procedure, would be suitable alternatives.

Both procedures described by Smith (1995) to carry out the automatic differentiation of the Cholesky factor of a matrix were implemented. Forward differentiation was set up to evaluate $\mathbf{L}$ and all its first and second derivatives to be as efficient as possible, ie, holding all matrices of derivatives in core and evaluating them simultaneously. In addition, it was set up to reduce memory requirements, ie, calculating the Cholesky factor and its first derivatives together and subsequently evaluating second derivatives individually. Backward differentiation was implemented storing all first derivatives of $\mathbf{M}$ in core but setting up and processing one second derivative of $\mathbf{M}$ at a time.

EXAMPLES

To illustrate the calculations, consider the test data for a bivariate analysis given by Meyer (1991). Fitting a simple animal model, there are six parameters. Table III summarises intermediate results for the likelihood and its derivatives for the starting values used by Meyer (1991), and gives estimates from the first round of iteration for simple or modified NR or MSC algorithms. Without reparameterisation, the (unmodified) NR algorithm produced estimates out of bounds of the parameter space, while the MSC performed quite well for this first iterate. Continuing to use expected values of second derivatives though, estimates failed to converge.

Figure 1 shows the corresponding change in log $\mathcal{L}$ over rounds of iteration. For this small example, with a starting value for the additive genetic covariance (σ_{A12}) very different from the eventual estimate, a NR or MSC algorithm optimising the step size (see equation [47]) failed to locate a suitable step size (which increased log $\mathcal{L}$) in the first iterate. Essentially, all algorithms had reached about the same point on the likelihood surface by the fifth iterate, with very little changes in log $\mathcal{L}$ in subsequent rounds. Convergence was considered achieved when the norm of the gradient vector was less than 10^{-4}. This was a rather strict criterion: changes in estimates and likelihood values between iterates were usually very small before it was met. Using Marquardt's (1963) modification (see [48]), six iterates plus 25

Table III. First and second derivatives of the log likelihood for the small numerical example (see Meyer (1991)), and estimates from the first round of iteration using NR and MSC algorithms.

	Parameter θ_i^a						log $\mathcal{L}^b$
	σ^2_{A1}	σ^2_{A12}	σ^2_{A2}	σ^2_{E1}	σ^2_{E12}	σ^2_{E2}	*+110*
Starting value	4.700	4.000	8.300	2.500	3.000	12.900	−72.3761
Derivatives of the likelihood							
$\partial \log \lvert \mathbf{G} \rvert / \partial \theta^c$	118.674	−114.385	67.201	0	0	0	
$\partial \log \lvert \mathbf{R} \rvert / \partial \theta_i^c$	0	0	0	157.574	−73.290	30.538	
$\partial \mathbf{y}'\mathbf{P}\mathbf{y} / \partial \theta_i^c$	−67.788	62.125	−22.640	−115.020	92.138	−32.634	
$\partial \log \mathcal{L} / \partial \theta_i^d$	12.711	−17.785	4.657	18.181	−23.182	5.264	
$\partial^2 \log \mathcal{L} / \partial \theta_i \partial \theta_i^e$	−9.3916	−10.9273	−1.3917	−24.4056	−19.0881	−1.7857	
$E[\partial^2 \log \mathcal{L} / \partial \theta_i \partial \theta_i]^f$	3.6515	3.2808	0.4420	12.2072	8.6807	0.9244	
$\partial^2 \log \mathcal{L} / \partial \theta_i \partial \theta_j^g$							
$j = 1$		−2.4029	0.4066	5.4955	−3.2745	0.4917	
$j = 2$	8.6021		−0.8456	−3.2745	4.1865	−0.9660	
$j = 3$	−1.9306	3.3691		0.4917	−0.9660	0.4755	
$j = 4$	−10.1028	7.9410	−1.5050		−6.9790	1.0028	
$j = 5$	7.9410	−9.1877	2.4708	17.6039		−1.9240	
$j = 6$	−1.5050	2.4708	−0.9674	−3.0447	4.7617		
Estimates on original scale							
NR$^{\tau^h=0}$, $\widehat{\theta}_i$	1.374	−2.155	0.347	2.936	2.811	13.195	−51.6973
MSC$^{\tau=0}$, $\widehat{\theta}_i$	4.214	−1.307	9.225	2.701	2.828	12.253	−43.1769
Estimates using reparameterisation							
ν_i^i	0.5098	1.3884	1.0581	0.2945	0.8353	1.2788	
$\partial \log \mathcal{L} / \partial \nu_i$	70.477	−15.943	6.173	65.535	−52.890	66.268	
NR$^{\tau=0}$							
$\widehat{\nu}_i$	0.3207	1.0759	0.7532	0.5717	0.2618	1.321	
$\widehat{\theta}_i$	3.057	2.285	4.510	3.206	0.983	14.101	−53.5470
NR$^{\tau=0.538}$							
$\widehat{\nu}_i$	0.7501	1.2420	0.7988	0.2087	0.4360	1.3395	
$\widehat{\theta}_i$	6.025	2.761	4.941	1.708	1.664	14.569	−49.3741
MSC$^{\tau=0}$							
$\widehat{\nu}_i$	0.8006	0.9098	0.9563	0.8727	−0.3909	1.5743	
$\widehat{\theta}_i$	5.786	2.367	6.770	5.881	−1.887	23.302	−84.8687
$\widehat{\theta}_i$	5.483	3.475	10.383	2.927	2.550	13.389	−49.4692
Converged estimates							
$\widehat{\theta}_i$	4.374	0.149	7.914	2.617	2.069	13.090	−42.0163
se $(\widehat{\theta}_i)^j$	1.026	1.292	3.001	0.567	0.809	2.073	

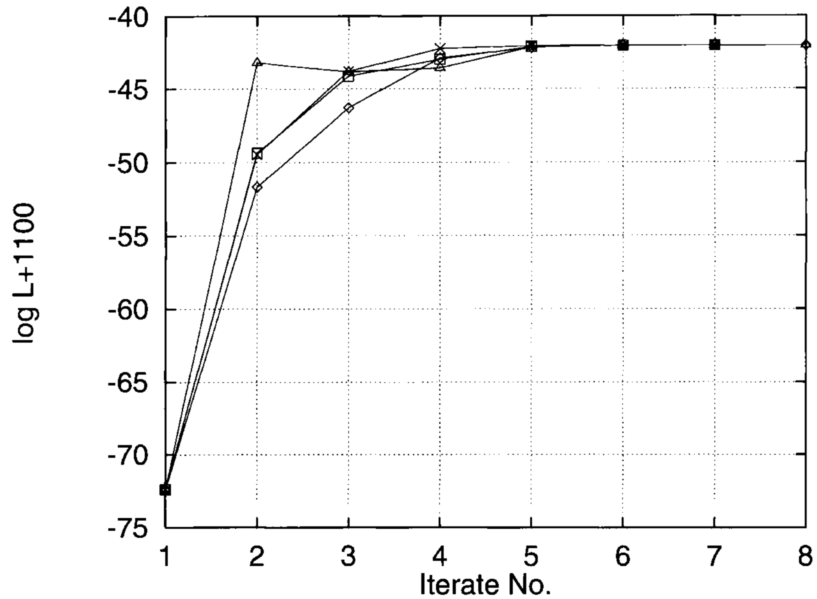

Fig 1. Change in log likelihood (L) over iterates for numerical example. ◇Newton–Raphson algorithm, original scale, unmodified; △: as ◇, but using Method of Scoring until change in log L is less than 1; □: Newton–Raphson algorithm with reparameterisation and modifying diagonals of the Hessian matrix; ×: as □, but using Method of Scoring until change in log L is less than 1.

likelihood evaluations were required to reach convergence using the observed information compared to eight iterates and 34 likelihood evaluations for the average information while the MSC (expected information) failed in this case. Optimising the step size (see [47]), $7+40$, $10+62$ and $10+57$ iterates + likelihood evaluations were required using the observed, expected and average information, respectively.

As illustrated in Figure 2, use of the 'average information' yielded a very similar convergence pattern to that of the NR algorithm. In comparison, using an EM algorithm for this example, 80 rounds of iteration were required before the change in log $\mathcal{L}$ between iterates was less than 10^{-4} and 142 iterates were needed before the average changes in estimates was less than 0.01%.

[a] σ_{Aij}: additive genetic covariances; $\sigma_{E_{ij}}$: residual covariances; [b] log likelihood: values given differ from those given by Meyer (1991) by a constant offset of 3.466 due to absorption of single-link parents without records ('pruning'). [c] Components of first derivatives of log $\mathcal{L}$; see text for definition. [d] First derivative of log $\mathcal{L}$ with respect to θ_i. [e] Second derivative of log $\mathcal{L}$ with respect to θ_i. [f] Expected value of second derivative of log $\mathcal{L}$ with respect to θ_i. [g] Second derivatives of log $\mathcal{L}$ with respect to θ_i and θ_j: observed values below, expected values above diagonal. [h] Modification factor for diagonals of matrix of second derivatives; see [48] in text. [i] Parameter on transformed scale. [j] Asymptotic lower bound sampling error, derived from inverse of observed information matrix.

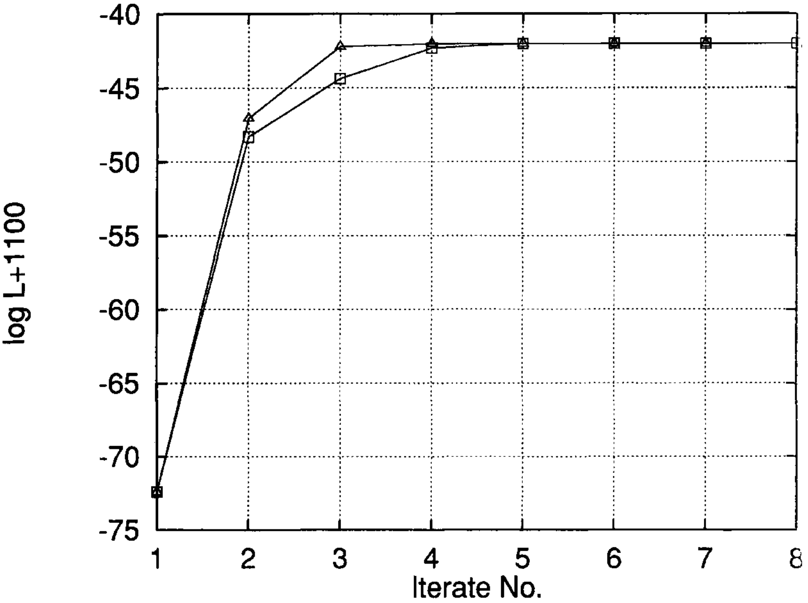

Fig 2. Change in log likelihood (L) over iterates for numerical example using 'average information' REML. △: Newton–Raphson algorithm, original scale, unmodified; □: Newton–Raphson algorithm with reparameterisation and modifying diagonals of the Hessian matrix.

Table IV gives characteristics of the data structure and model of analysis for applied examples of multivariate analyses of beef cattle data. The first is a bivariate analysis fitting an animal model with maternal permanent environmental effects as an additional random effect, ie, estimating nine covariance components. The second shows the analysis of three weight traits in Zebu Cross cattle (analysed previously; see Meyer (1994a)) under three models of analysis, estimating up to 24 parameters.

For each data set and model, the computational requirements to obtain derivatives of the likelihood were determined using forward and backward differentiation

[a] NR: Newton-Raphson; MSC: method of scoring; MIX: Starting as MSC, switching to NR when change in log likelihood between iterates drops below 1.0; and DF: derivative-free algorithm. [b] CB: Cannon bone length; HH: hip height; WW: weaning weight; YW: yearling weight; FW: final weight. [c] 1: simple animal model; 2: animal model fitting dams' permanent environmental effect; and 5: animal model fitting both genetic and permanent environmental effects. [d] A: Forward differentiation, processing all derivatives simultaneously; B: Forward differentiation, processing second derivatives one at a time; C: Backward differentation. [e] 0: No reparameterisation, unmodified; S: reparameterised scale, optimising step size, see [47] in text; T: reparameterised scale, modifying diagonals of matrix of second derivatives, see [48] in text. [f] Using backward differentiation to obtain derivatives.

Table IV. Characteristics of the data structure, model of analysis and convergence for analyses of beef data sets using different algorithms[a] and methods to calculate derivatives of the likelihood.

	Trait[b]			
	CB + HH		*WW + YW + FW*	
	Hereford		*Zebu Crosses*	
	Model[c] *2*	*Model*[c] *1*	*Model*[c] *2*	*Model*[c] *5*
Characteristics of the analysis				
No parameters	9	12	18	24
No 2nd derivatives	45	78	171	300
No records	2 664	7 443	7 443	7 443
No animals	2 369	3 648	3 648	3 648
No of rows in **M**	6 193	10 231	14 275	25 990
No of elements in **L**	258 764	391 288	533 066	1 199 206
Time (s) per evaluation				
$\log \mathcal{L}$ only	6.5	6.8	8.4	41.7
$\log \mathcal{L}$ + derivatives[d]				
A	371	–	–	–
B	531	846	2 134	21 215
C	216	364	731	4 797
No of iterates + likelihood evaluations				
NR				
0[e]	5 + 10	–	–	–
S	6 + 48	8 + 48	(9 + 64)	11 + 66
T	(20 + 135)	9 + 17	16 + 33	13 + 21
MSC				
0	7 + 0	–	–	–
S	7 + 36	–	–	–
T	8 + 20	–	–	–
MIX				
S	5 + 29	Fail	12 + 83	–
T	6 + 18	11 + 18	18 + 26	–
DF	0 + 699	0 + 1236	0 + 4751	
Time (mins) per analysis[f]				
NR				
0	18	–	–	–
S	27	54	(119)	925
T	(87)	57	200	1 054
MSC				
0	25	–	–	–
S	29	–	–	–
T	31	–	–	–
MIX				
S	21	Fail	158	–
T	24	69	223	–
DF	76	140	665	

as described by Smith (1995). If the memory available allowed, forward differentiation was carried out for all derivatives simultaneously and considering one matrix $\partial^2 \mathbf{M}/\partial\theta_i\partial\theta_j$ at a time. Table IV gives computing times in minutes for calculations carried out on a DEC Alpha Chip (DIGITAL) machine running under OSF/1 (rated at about 200 Mflops). Clearly, the backward differentiation is most competitive, with the time required to calculate 24 first and 300 second derivatives being 'only' about 120 times that to obtain log $\mathcal{L}$ only.

Starting values used were usually 'good', using estimates of variance components from corresponding univariate analyses throughout and deriving initial guesses for covariance components from these and literature values for the correlations concerned. A maximum of 20 iterates was carried out, applying a very stringent convergence criterion of a change in log $\mathcal{L}$ less than 10^{-6} between iterates or a value for the norm of the gradient vector of less than 10^{-4} as above.

The numbers of iterates and additional likelihood evaluations required for each algorithm given in table IV show small and inconsistent differences between them. On the whole, starting with a MSC algorithm and switching to NR when the change in log $\mathcal{L}$ became small and applying Marquardt's (1963) modification (see [48]) tended to be more robust (ie, achieve convergence when other algorithm(s) failed) for starting values not too close to the eventual estimates. Conversely, they tended to be slower to converge than an NR optimising step size (see [47]) for starting values very close to the estimates. However, once the derivatives of log $\mathcal{L}$ have been evaluated, it is computationally relatively inexpensive (requiring only simple likelihood evaluations) to compare and switch between algorithms, ie, it should be feasible to select the best procedure to be used for each analysis individually.

Comparisons with EM algorithms were not carried out for these examples. However, Misztal (1994) claimed that each sparse matrix inversion required for an EM step took about three times as long as one likelihood evaluation. This would mean that each iterate using second derivatives for the three-trait analysis estimating 24 highly correlated (co)variance components would require about the same time as 40 EM iterates, or that the second derivative algorithm would be advantageous if the EM algorithm required more than 462 iterates to converge.

DISCUSSION

For univariate analyses, Meyer (1994b) found no advantage in using Newton-type algorithms over derivative-free REML. However, comparing total cpu time per analysis, using derivatives appears to be highly advantageous over a derivative-free algorithm, for multivariate analyses, the more so the larger the number of parameters to be estimated. Furthermore, for analyses involving larger numbers of parameters (18 or 24), the derivative-free algorithm converged several times to a local maximum, a problem observed previously by Groeneveld and Kovac (1990), while the Newton-type algorithm appeared not be affected.

Modifications of the NR algorithm, combined with a local search for the optimum step size, improved its convergence rate. This was achieved primarily by safeguarding against steps in the 'wrong' direction in early iterates, thus making the search for the maximum of the likelihood function more robust against bad starting values. With likelihood evaluations requiring only a small fraction of the time required per NR iterate, this generally resulted in reduction in the total cpu time required per analysis. Similarly, a reparameterisation removed constraints on the parameters, and thus reduced the incidence of failure to converge because estimates were out of the parameter space. However, for cases where maximisation on the original scale was successful, this tended to increase the number of iterates required.

Recently, the rediscovery of an efficient algorithm to invert a large sparse matrix (Takahashi et al, 1973) has made the use of algorithms which require the direct inverse of the coefficient matrix in the mixed-model equations feasible for large animal model analyses. Hence we now have a range of REML algorithms of increasing complexity available. These range from derivative-free procedures to first derivative methods (with or without the approximation of second derivatives, either by finite differences or consecutive updates) to algorithms for which second derivatives of the likelihood (observed, expected or their average) are calculated. Each has its particular computational requirements, in terms of cpu time and memory required, and convergence behaviour. There is no globally 'best' method; the choice in a particular case is determined by the resources available, model to be fitted and size of the data set to be analysed.

On the whole, the more parameters are to be estimated and the stronger sampling correlations between parameters are (eg, for models including direct and maternal effects and direct-maternal covariances), the more advantageous second derivative methods tend to become.

CONCLUSIONS

Evaluation of derivatives of the likelihood for animal models without the need to invert large matrices is feasible. This is achieved through a straight extension of the methodology applied in calculating the likelihood only. Smith's (1995) automatic differentiation procedure adds a valuable tool to the numerical procedures available.

A simple reparameterisation transforms the constrained maximisation problem posed in the estimation of variance components to an unconstrained one. This allows the use of an (extended) NR algorithm to locate the maximum of the likelihood function. It is equally useful for Quasi-Newton algorithms approximating second derivatives.

Experience so far has shown second derivative algorithms to be highly advantageous over derivative-free procedures for multiparameter problems, in terms of both computing time required and robustness against convergence to local maxima. Savings in computing time, however, are obtained at the expense of extra memory required.

ACKNOWLEDGMENT

This work was supported by grant UNE35 from the Australian Meat Research Corporation.

REFERENCES

Boldman KG, Van Vleck LD (1991) Derivative-free restricted maximum likelihood estimation in animal models with a sparse matrix solver. *J Dairy Sci* 74 4337–4343

Box MJ (1965) A comparison of several current optimization methods, and the use of transformations in constrained problems. *Computer J* 9 67–77

Dennis JE Jr, Moré JJ (1977) Quasi-Newton methods, motivation and theory. *SIAM Rev* 19, 46–89

Ducrocq V (1993) Genetic parameters for type traits in the French Holstein breed based on a multiple-trait animal model. *Livest Prod Sci*, 36, 143–156

Gill PE, Murray W, Wright MH (1981) *Practical Optimization.* Academic Press, New York

George A, Liu JWH (1981) *Computer Solution of Large Positive Definite Systems.* Prentice-Hall Inc, Englewood Cliffs, NJ

Graser HU, Smith SP, Tier B (1987) A derivative-free approach for estimating variance components in animal models by restricted maximum likelihood. *J Anim Sci* 64, 1362–1370

Groeneveld E (1994) A reparameterisation to improve numerical optimization in multivariate REML (co)variance component estimation. *Genet Select Evol* 26, 537–545

Groeneveld E, Kovac M (1990) A note on multiple solutions in multivariate restricted maximum likelihood covariance component estimation. *J Dairy Sci* 73, 2221–2229

Groeneveld E, Lacher P, Kovac M (1991) Simultaneous estimation of 60 covariance components using a derivative-free multivariate REML procedure. *J Anim Sci* 69, Suppl 1, 189 (Abstr)

Harville DA (1977) Maximum likelihood approaches to variance component estimation and related problems. *J Am Stat Ass* 72, 320–338

Harville DA, Callanan TP (1990) Computational aspects of likelihood-based inference for variance components. In: *Proc Int Symp on Adv Stat Methods Genet Improvement Livest* (D Gianola, K Hammond, eds), Springer Verlag, Heidelberg

Hill WG, Thompson R (1978) Probabilities of non-positive between-group or genetic covariance matrices. *Biometrics* 34, 429–439

Jennrich RI, Sampson PF (1976) Newton-Raphson and related algorithms for maximum likelihood variance component estimation. *Technometrics* 18, 11–17

Jennrich RI, Schluchter MD (1986) Unbalanced repeated measures models with structural covariance matrices. *Biometrics* 42, 805–820

Jensen J, Mao IL (1988) Transformation algorithms in analysis of single trait and of multitrait models with equal design matrices and one random factor per trait; a review. *J Anim Sci* 66, 2750–2761

Jensen ST, Johansen S, Lauritzen SL (1991) Globally convergent algorithms for maximizing a likelihood function. *Biometrika* 78, 867–877

Johnson DL, Thompson R (1995) Restricted maximum likelihood estimation of variance components for univariate animal models using sparse matrix techniques and average information. *J Dairy Sci* 78, 449–456

Lindstrom MJ, Bates DM (1988) Newton–Raphson and EM algorithms for linear mixed-effects models for repeated-measures data. *J Am Stat Ass* 83, 1014–1022

Lindstrom MJ, Bates DM (1994) Corrections to Lindstrom and Bates (1988). *J Am Stat Ass* 89, 1572

Marquardt DW (1963) An algorithm for least squares estimation of nonlinear parameters. *SIAM J* 11, 431–441

Meyer K (1985) Maximum likelihood estimation of variance components for a multivariate mixed model with equal design matrices. *Biometrics* 41, 153–166

Meyer K (1989) Restricted maximum likelihood to estimate variance components for animal models with several random effects using a derivative-free algorithm. *Genet Select Evol* 21, 317–340

Meyer K (1991) Estimating variances and covariances for multivariate animal models by restricted maximum Likelihood. *Genet Select Evol* 23, 67–83

Meyer K (1992) *DFREML Version 2.1 – Programs to Estimate Variance Components by Restricted Maximum Likelihood Using a Derivative-Free Algorithm. User Notes.* AGBU, University of New England, Armidale NSW, Mimeo, 101 p

Meyer K (1994a) Estimates of direct and maternal correlations among growth traits in Australian beef cattle. *Livest Prod Sci* 38, 91–105

Meyer K (1994b) Derivative-intense restricted maximum likelihood estimation of covariance components for animal models. *World Congr Genet Appl Livest Prod* Vol 18 (C Smith, JS Gavona, B Benkel, J Chesnais, W Fairfull, J Gibson, BW Kennedy, EB Burnside, eds), Univ of Guelph, Guelph, ON, Canada, 365–369

Misztal I (1990) Restricted maximum likelihood estimation of variance components in animal models using sparse matrix inversion and a supercomputer. *J Dairy Sci* 73, 163–172

Misztal I (1994) Comparison of computing properties of derivative and derivative-free algorithms in variance component estimation by REML. *J Anim Breed Genet* 111, 346–355

Patterson HD, Thompson R (1971) Recovery of inter-block information when block sizes are unequal. *Biometrika* 58, 545–554

Powell MJD (1970) A survey of numerical methods for unconstrained optimization. *Siam Rev* 12, 79–97

Searle SR (1982) *Matrix Algebra Useful for Statistic.* John Wiley & Sons, New York

Searle SR, Casella G, McCulloch CE (1992) *Variance Components.* John Wiley & Sons, New York

Smith SP (1995) Differentiation of the Cholesky algorithm. *J Comp Graph Stat* 4, 134–147

Takahashi K, Fagan J, Chen M (1973) Formation of a sparse bus impendance matrix and its application to short circuit study. In: *8th Power Industry Computer Applications Conference*, New York, IEEE, 63–69; cited by Smith (1995)

Tier B, Smith SP (1989) Use of sparse matrix absorption in animal breeding. *Genet Select Evol* 21, 457–466

Thompson R (1973) The estimation of variance, covariance components with an application when records are subject to culling. *Biometrics* 29, 527-550

Thompson R, Meyer K (1986) Estimation of variance components: what is missing in the EM algorithm? *J Stat Comp Simul* 24, 215–230

Zacks S (1971) *The Theory of Statistical Inference.* John Wiley & Sons, New York